**C# for loop**

In this article, we will learn about for loop in C# and different ways to use them in a program.

In programming, it is often desired to execute certain block of statements for a specified number of times. A possible solution will be to type those statements for the required number of times. However, the number of repetitions may not be known in advance (during compile time) or maybe large enough (say 10000).

The best solution to such problem is loop. Loops are used in programming to repeatedly execute a certain block of statements until some condition is met.

In this article, we’ll look at for loop in C#.

**C# for loop**

The **for** keyword is used to create for loop in C#. The syntax for **for loop** is:

for (initialization; condition; iterator)

{

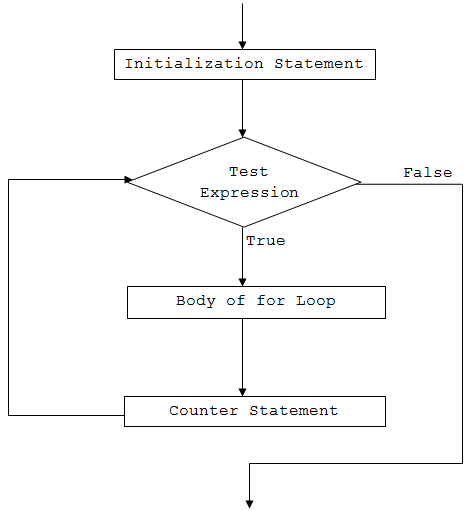
// body of for loop

}

**How for loop works?**

1. C# for loop has three statements: initialization, condition and iterator.
2. The initialization statement is executed at first and only once. Here, the variable is usually declared and initialized.
3. Then, the condition is evaluated. The condition is a boolean expression, i.e. it returns either true or false.
4. If the condition is evaluated to true:
   1. The statements inside the for loop are executed.
   2. Then, the iterator statement is executed which usually changes the value of the initialized variable.
   3. Again, the condition is evaluated.
   4. The process continues until the condition is evaluated to false.
5. If the condition is evaluated to false, the for loop terminates.

**for Loop Flowchart**

Working of C# for loop

**Example 1: C# for Loop**

using System;

namespace Loop

{

class ForLoop

{

public static void Main(string[] args)

{

for (int i=1; i<=5; i++)

{

Console.WriteLine("C# For Loop: Iteration {0}", i);

}

}

}

}

When we run the program, the output will be:

C# For Loop: Iteration 1

C# For Loop: Iteration 2

C# For Loop: Iteration 3

C# For Loop: Iteration 4

C# For Loop: Iteration 5

In this program,

* initialization statement is int i=1
* condition statement is i<=5
* iterator statement is i++

When the program runs,

* First, the variable i is declared and initialized to 1.
* Then, the condition (i<=5) is evaluated.
* Since, the condition returns true, the program then executes the body of the for loop. It prints the given line with Iteration 1 (Iteration simply means repetition).
* Now, the iterator (i++) is evaluated. This increments the value of i to 2.
* The condition (i<=5) is evaluated again and at the end, the value of i is incremented by 1. The condition will evaluate to true for the first 5 times.
* When the value of i will be 6 and the condition will be false, hence the loop will terminate.

**Example 2: for loop to compute sum of first n natural numbers**

using System;

namespace Loop

{

class ForLoop

{

public static void Main(string[] args)

{

int n = 5,sum = 0;

for (int i=1; i<=n; i++)

{

// sum = sum + i;

sum += i;

}

Console.WriteLine("Sum of first {0} natural numbers = {1}", n, sum);

}

}

}

When we run the program, the output will be:

Sum of first 5 natural numbers = 15

Here, the value of sum and n are initialized to 0 and 5 respectively. The iteration variable i is initialized to 1 and incremented on each iteration.

Inside the for loop, value of sum is incremented by i i.e. sum = sum + i. The for loop continues until i is less than or equal to n (user's input).

Let's see what happens in the given program on each iteration.

Initially, i = 1, sum = 0 and n = 3

| For loop execution steps | | | |
| --- | --- | --- | --- |
| Iteration | Value of i | i<=5 | Value of sum |
| 1 | 1 | True | 0+1 = 1 |
| 2 | 2 | True | 1+2 = 3 |
| 3 | 3 | True | 3+3 = 6 |
| 4 | 4 | True | 6+4 = 10 |
| 5 | 5 | True | 10+5 = 15 |
| 6 | 6 | False | Loop terminates |

So, the final value of sum will be 15 when n = 5.

**Multiple expressions inside a for loop**

We can also use multiple expressions inside a for loop. It means we can have more than one initialization and/or iterator statements within a for loop. Let's see the example below.

**Example 3: for loop with multiple initialization and iterator expressions**

using System;

namespace Loop

{

class ForLoop

{

public static void Main(string[] args)

{

for (int i=0, j=0; i+j<=5; i++, j++)

{

Console.WriteLine("i = {0} and j = {1}", i,j);

}

}

}

}

When we run the program, the output will be:

i = 0 and j = 0

i = 1 and j = 1

i = 2 and j = 2

In this program, we have declared and initialized two variables: i and j in the initialization statement.

Also, we have two expressions in the iterator part. That means both i and j are incremented by 1 on each iteration.

**For loop without initialization and iterator statements**

The initialization, condition and the iterator statement are optional in a for loop. It means we can run a for loop without these statements as well.

In such cases, for loop acts as a [while loop](https://www.programiz.com/csharp-programming/do-while-loop). Let's see the example below.

**Example 4: for loop without initialization and iterator statement**

using System;

namespace Loop

{

class ForLoop

{

public static void Main(string[] args)

{

int i = 1;

for ( ; i<=5; )

{

Console.WriteLine("C# For Loop: Iteration {0}", i);

i++;

}

}

}

}

When we run the program, the output will be:

C# For Loop: Iteration 1

C# For Loop: Iteration 2

C# For Loop: Iteration 3

C# For Loop: Iteration 4

C# For Loop: Iteration 5

In this example, we haven't used the initialization and iterator statement.

The variable i is initialized above the for loop and its value is incremented inside the body of loop. This program is same as the one in Example 1.

Similarly, the condition is also an optional statement. However, if we don't use test expression, the for loop won't test any condition and will run forever (infinite loop).

**Infinite for loop**

If the condition in a for loop is always true, for loop will run forever. This is called infinite for loop.

**Example 5: Infinite for loop**

using System;

namespace Loop

{

class ForLoop

{

public static void Main(string[] args)

{

for (int i=1; i>0; i++)

{

Console.WriteLine("C# For Loop: Iteration {0}", i);

}

}

}

}

Here, i is initialized to 1 and the condition is i>0. On each iteration we are incrementing the value of i by 1, so the condition will never be false. This will cause the loop to execute infinitely.

We can also create an infinite loop by replacing the condition with a blank. For example,

for ( ; ; )

{

// body of for loop

}

or

for (initialization; ; iterator)

{

// body of for loop

}